



Project Scheduling: PERT/CPM

Project Scheduling with Known Activity Times (as in exercises 1, 2, 3 and 5 in the handout) and considering Time-Cost Trade-Offs (as in exercises 4 and 6 in the handout)

This is ***Critical Path Method***

Project Scheduling with Uncertain Activity Times (as in exercises 7, 8 and 9 in the handout).

This is ***Program Evaluation Research Technique***

Acknowledgement: I have adapted the format and some ideas from the PowerPoint slides of Dr. C. Lightner of Fayetteville State University, but most of the mathematical solutions and method I have used are quite different and extend his work.



Introduction to Project Management

- ⊕ Project Scheduling or project management is used to schedule, manage and control projects which can be analyzed into various semi-independent activities or tasks.

- ⊕ Example: Building a New Home
When building a home individual subcontractors are hired to:
 - Grade and prepare the land
 - Build the foundation
 - Frame up the home
 - Insulate the home
 - Wire (Electricity, Cable, Telephone lines) the home
 - Drywall
 - Paint (inside)
 - Put vinyl siding on home
 - Install Carpet
 - Landscape
 - Lay Concrete



PERT/CPM

⊕ PERT

- Program Evaluation and Review Technique
- Developed by U.S. Navy for Polaris missile project
- Developed to handle uncertain activity times

⊕ CPM

- Critical Path Method
- Developed by Du Pont & Remington Rand
- Developed for industrial projects for which activity times are known

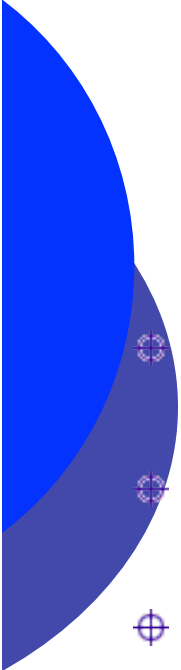
- ⊕ There are project management software packages that can perform both.



PERT/CPM

- ⊕ PERT and CPM have been used to plan, schedule, and control a wide variety of projects:
 - R&D of new products and processes
 - Construction of buildings and highways
 - Maintenance of large and complex equipment
 - Design and installation of management systems
 - Organizing transportation projects
 - Deployment and/or relocation of forces
 - Design of computer systems

PERT/CPM

- 
- ⊕ PERT/CPM is used to plan the scheduling and optimal staffing of individual [activities](#) that make up a project.
 - ⊕ Projects may have as many as several thousand activities and may have to be broken up into simpler sub-projects.
 - ⊕ Usually some activities depend on the completion of other activities before they can be started.
 - ⊕ So we need to start with the [Prerequisites Task Set](#) giving the order of precedencies, along with durations for each task, or activity



PERT/CPM

Project managers rely on PERT/CPM to help them answer questions such as:

- What is the total time to complete the project?
- What are the scheduled start and finish dates for each specific activity?
- Which activities are critical and must be completed exactly as scheduled to keep the project on schedule?
- How long can noncritical activities be delayed before they cause an increase in the project completion time?
- If activity times are uncertain (i.e., they are random variables), what is the probability that the project will be finished in any given time-frame?
- If there is a cost associated with “crashing” each activity, what is the extra cost of completing the project at a time earlier than the normal time?



Project Network

- ⊕ A project network can be constructed to model the precedence of the activities. It consists of nodes and arrows between the nodes.
- ⊕ Each activity is represented by an arrow, which starts at a node and ends at a different node.
- ⊕ The nodes of the network represent events comprising the completion of a subset of activities, so that subsequent activities can occur.
- ⊕ The start node has only arrows issuing from it; the finish node has only arrows entering it. All other nodes have arrows both entering and leaving.
- ⊕ Immediate predecessor(s) is (are) activities that must be completed immediately before a given activity can begin. The arrows entering a node represent the immediate predecessors of those exiting the node.
- ⊕ The network is a representation of the precedence relationships of the activities.



The Critical Path and Activity Slack

- ⊕ A path through a network is one of the routes following the arrows (arcs) from the start node to the finish node.
- ⊕ The length of a path is the *sum* of the (estimated) *durations* of the activities on the path.
- ⊕ The (estimated) project duration or project completion time equals the *length of the longest path* through the project network.
- ⊕ This longest path is called the critical path. (If more than one path tie for the longest, they are all critical paths.)
- ⊕ Activity Slack is the amount of time that noncritical activities can be delayed without increasing the project completion time.
- ⊕ The critical path will always consist of activities with zero slack.



Constructing Project Networks

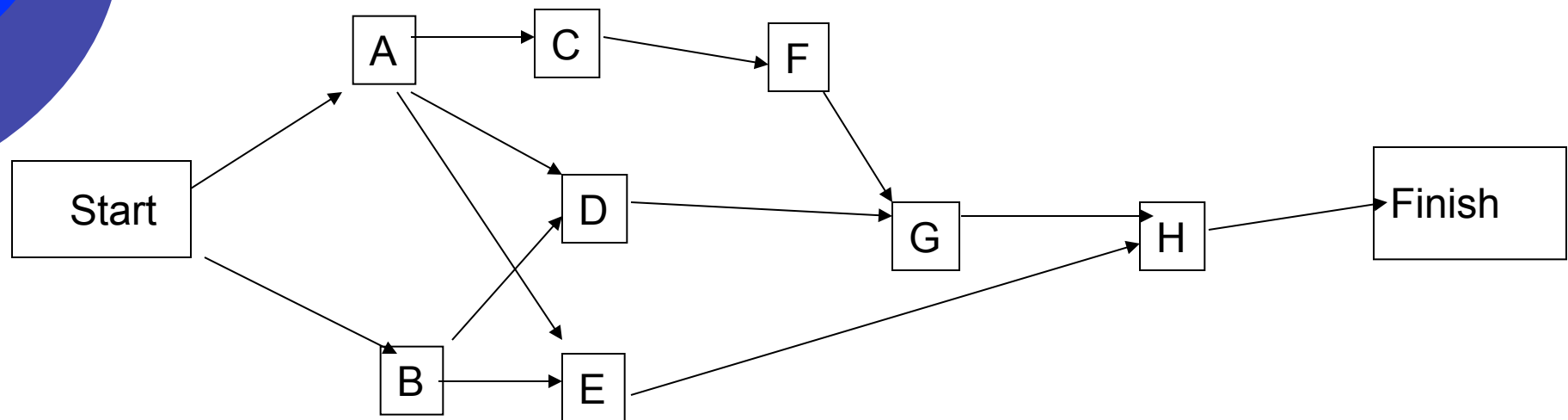
- ⊕ The Mohawk Discount Store is designing a management training program for individuals at its corporate headquarters. The company wants to design a program so that trainees can complete it as quickly as possible. Important precedence relationships must be maintained between assignments or activities in the program. For example, a trainee cannot serve as an assistant to the store manager until the employee has obtained experience in the credit department and at least one sales department. The following activities are the assignments that must be completed by each program trainee. Construct the project network for this problem. (Anderson, et. Al, Chapter 10, problem 1).

- ⊕ Try this yourself before looking at the next slide!

Activities A – H represent actual tasks.

Activity	A	B	C	D	E	F	G	H
Immediate Predecessor	---	---	A	A, B	A,B	C	D,F	E,G

Mohawk Project Network





Project Network Discussion

- ⊕ Project networks are not unique. A project network is considered valid provided all precedence relationships are preserved.
- ⊕ If it is convenient, you can introduce an activity with zero duration, a so-called [dummy activity](#), so as to preserve precedence relationships. Sometimes you can simplify a network by re-drawing it with the arrows in different positions (but with the same precedence relations). Try to minimize the use of dummy activities.
- ⊕ ONLY ACTIVITIES THAT ARE NOT IMMEDIATE PREDECESSORS TO ANY OTHER NETWORK ACTIVITIES MAY HAVE A LINK DIRECTLY TO THE FINISH NODE.
- ⊕ No dangling activities (i.e., with no beginning or end node) and no loops are allowed in the diagram.



Overall Procedure for solving a Project Network

1. Determine the sequence of activities.
2. Construct the network or precedence diagram. Make sure there are no “dangling” activities or loops: all activities lead towards the end of the project.
3. Starting from the left, compute the Early Start (**ES**) time at each node. This is the FORWARD PASS.
4. Starting from the right, compute the Late Finish (**LF**) time at each node. This is the BACKWARD PASS.
5. Find the slack(s) for each activity.
6. Identify the Critical Path.

In the following slides will elaborate on steps 3-6.

Project Management Notation: Nodes

ET

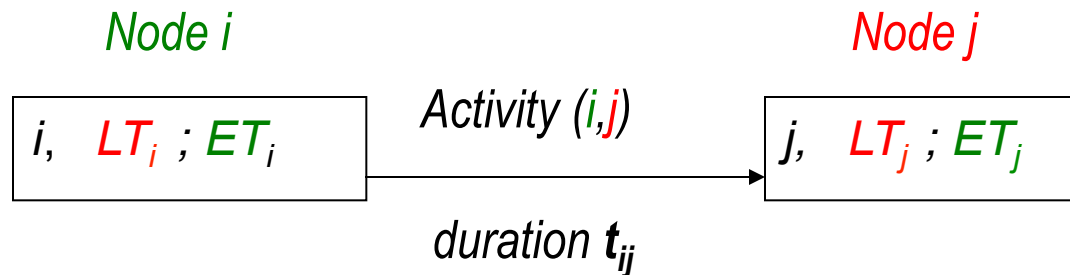
The earliest time *all* the activities entering a node can be said to be completed. This will be computed on a *forward pass* through the network, starting at the *start* node.

LT

The latest time the node at the end of the activity can be completed and not delay the project. This will be computed on a *backward pass* through the network, starting at the *finish* node.

t_{ij}

Duration of activity (*i,j*) (shorthand for the activity going from node *i* to node *j*).



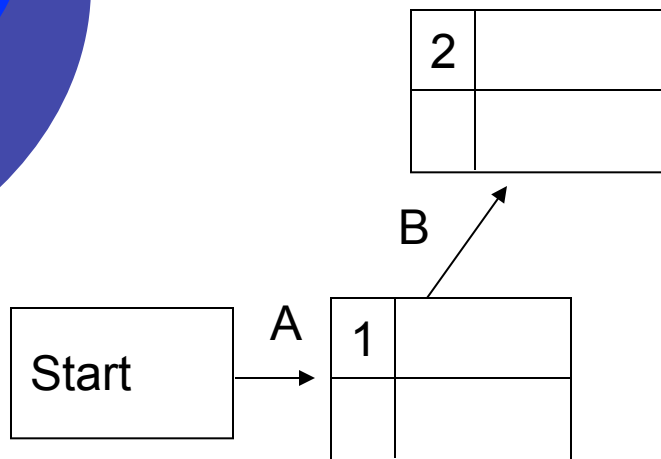


Example: Horizon Cable

- ⊕ Horizon is about to expand its cable TV offerings in Smalltown, adding the Cooking Network and others. The activities in the table below are needed to complete the service expansion.

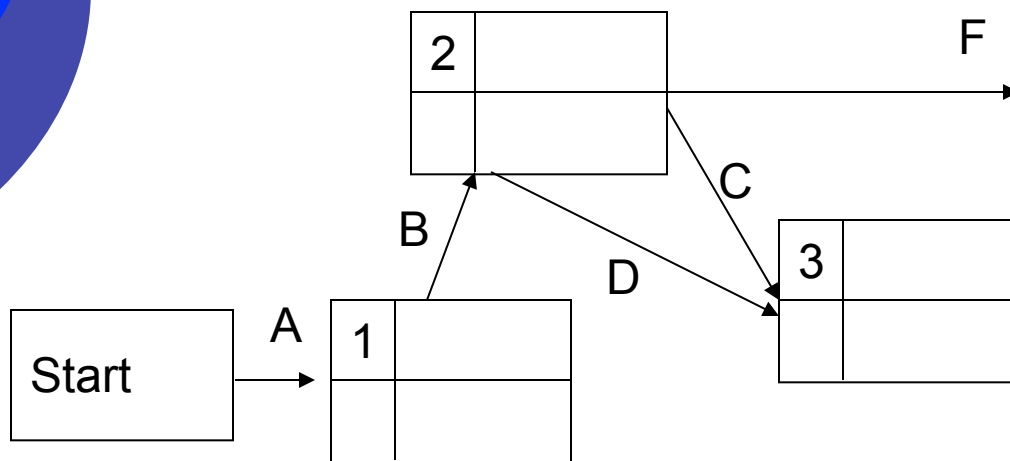
Activity	Description	Immediate Predecessors	Duration (weeks)
A	Choose stations	-	2
B	Get town council approval	A	4
C	Order converters	B	3
D	Install new dish to receive new stations	B	2
E	Install converters	C, D	10
F	Change billing system	B	4

Example: Horizon Cable



We put in the Start node on the left and see that activity A is the only one with no predecessors. This is the first arrow, leading to node 1. Then we see that the only activity with A as predecessor is B, so we attach an arrow representing B to Node 1, with Node 2 at its head.

Example: Horizon Cable

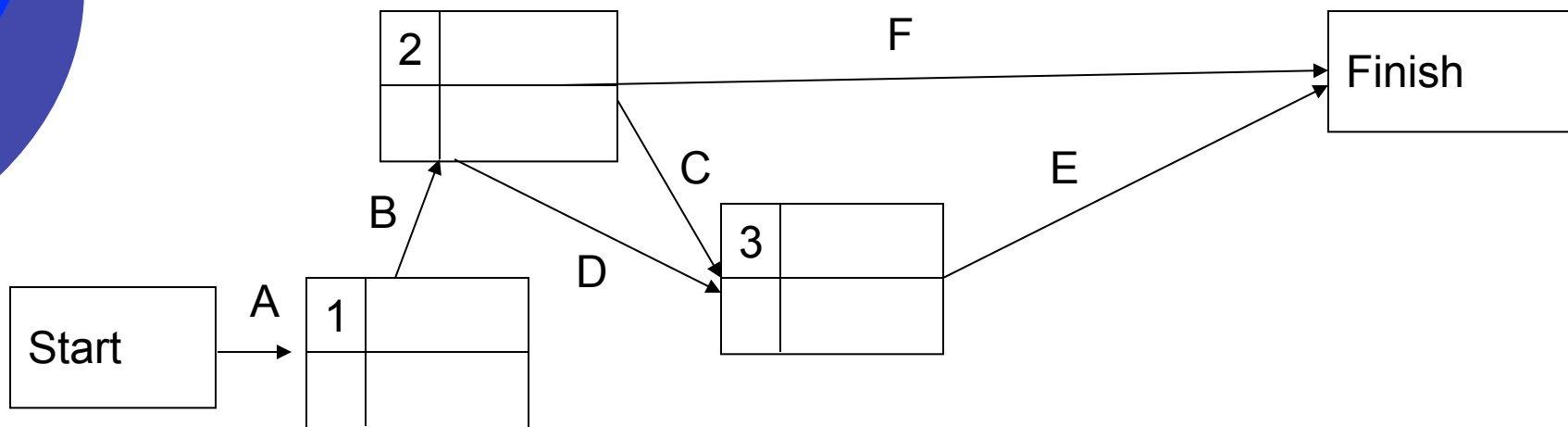


Activity	Immediate Predecessors
A	-
B	A
C	B
D	B
E	C, D
F	B

Now from the table of precedencies, we see that activities C, D and F are those that have B as an immediate predecessor, so we attach an arrow for each with its base at node 2. Activities C and D both end in a single node # 3– to the right of node 2. This is because a single activity, E, depends on the completion of both of them together.

F has no ending node just yet: no activity has F as a predecessor, so it will go to the End node.

Example: Horizon Cable



Activity	Immediate Predecessors
A	-
B	A
C	B
D	B
E	C, D
F	B

Since no activities depend for their start on E and F, both of these end in the Finish node and we are done drawing the precedence relationship as a network.

It's considered best to avoid having arrows crossing each other. While this is not exactly wrong, neither is it particularly aesthetic, so we try to avoid such crossings.

Earliest Node, Start and Finish Times

⊕ **Step 3:** Make a forward pass through the network as follows: For each node i beginning at the Start node (time 0), compute:

- Earliest Node Time **ET** for any node is the *maximum* of the earliest *finish* times of all activities immediately preceding activity i : an activity starts at its tail node's time: i.e., once *all* its predecessors have completed
- Earliest Finish Time **EFT** = (Earliest Tail Node i Time) + (Duration of activity ij).

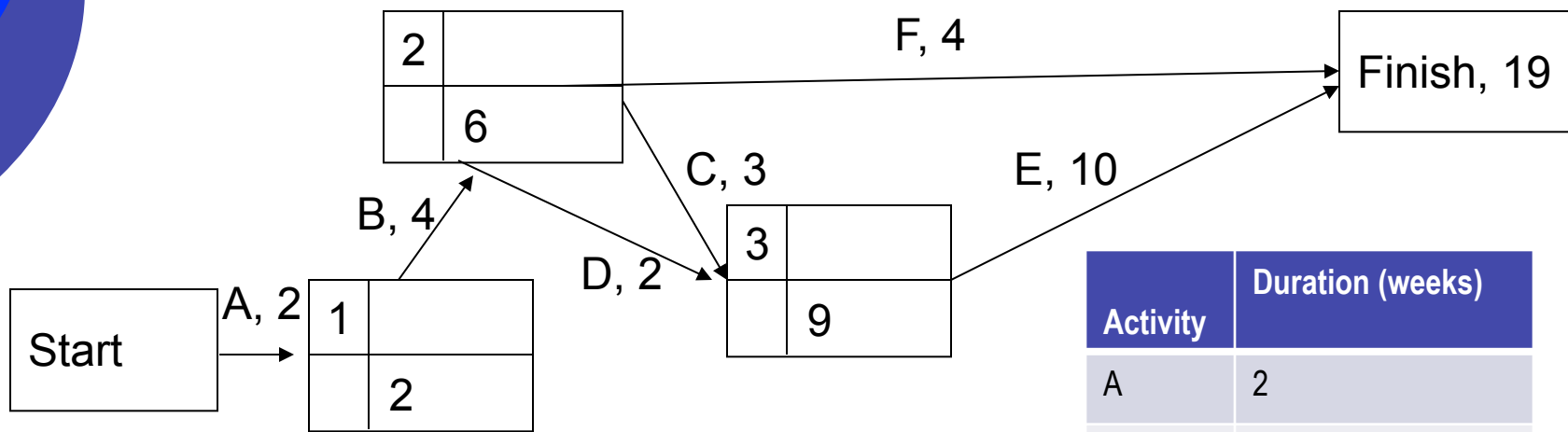
ET is 0 for nodes with no predecessors, such as the Start node.

Note: The project completion time is the maximum of the Earliest Finish Times at the Finish node.

In the following, each node represents the completion of an activity. The Earliest time for the node is written on the the lower right.

Node # _____
ET = max{ EFT (= EST +duration) for all predecessors}

Earliest Times: Horizon Cable



Activity	Duration (weeks)
A	2
B	4
C	3
D	2
E	10
F	4

So the total project time is 19 days.



Latest Node Times

Step 4: Make a *backwards* pass through the network, and at each node apply the **Latest Finish Time Rule**: $LT = \text{Minimum (LT-duration) of the immediate successor nodes}$ as follows:

(The immediate successors for a node are all nodes that are at the head of an arrow issuing from the current node.)

Procedure for obtaining latest times for all activities:

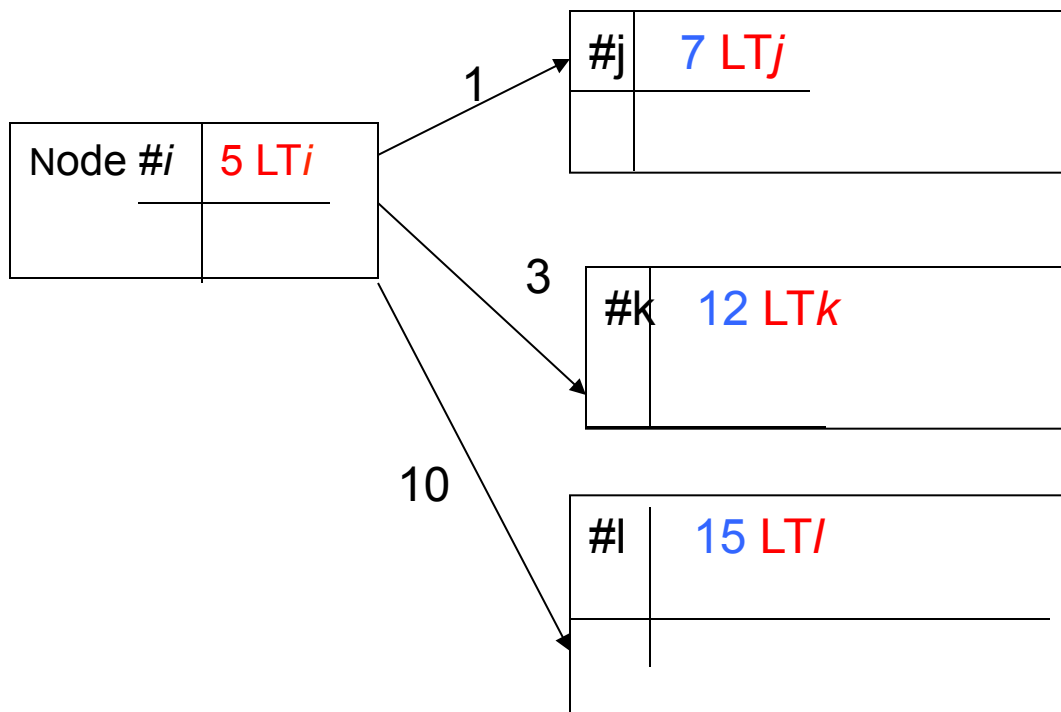
- For the finish node, set LT equal to project completion time.
- For each activity issuing from a given prior node, calculate (LT of its finish node) – (duration of the current activity)
- Apply the *latest finish time rule*.
- Repeat until LFT and LST have been obtained for *all* activities.

Note: The minimum of ST for nodes directly subsequent to the Start Node should be just the duration of the activity leading to them!

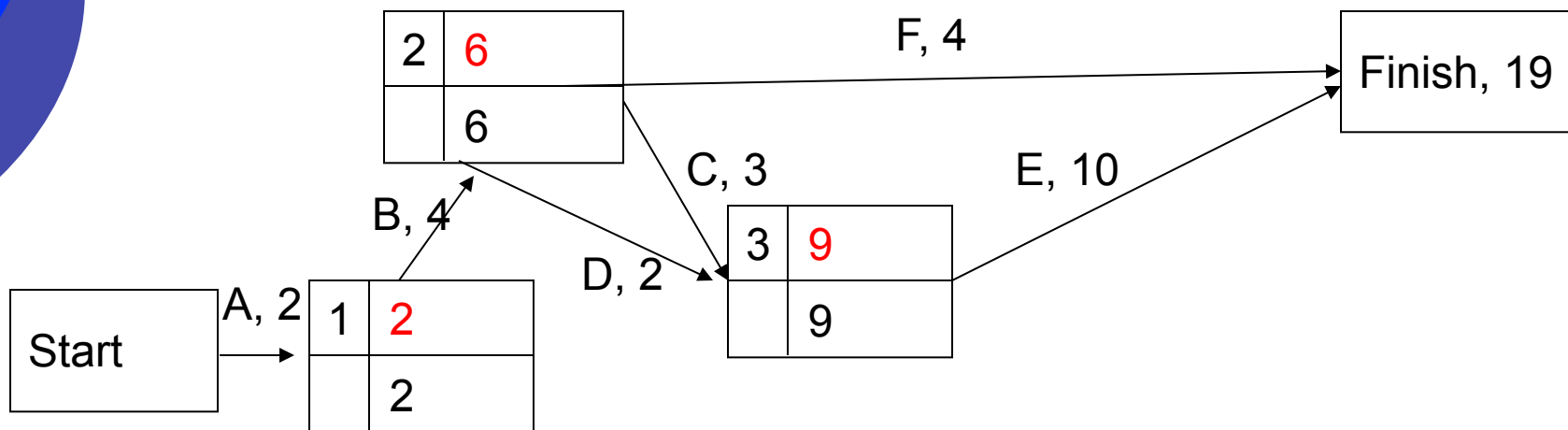
Finish Times

Node #	LT
	ET

E.g.: the red 5 is the minimum of the blue $7-1=6$, $12-3=9$ and $15-10=5$



Example: Horizon Cable



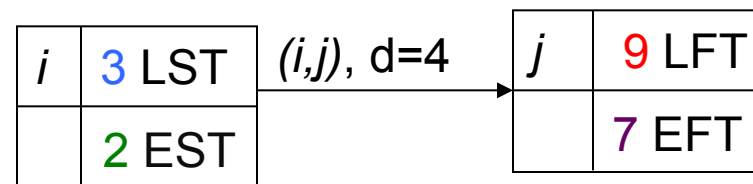
Note: The only path through the network with zero total slack activities (i.e., no leeway in start and finish times) is:

A – B – C – E

This is the **critical path**. There is always at least one critical path . In this example all nodes are zero slack also.

Activity Times and Slack

Each activity has a start node and an end node. In the example below, there are different earliest times and latest times in each of nodes i and j relevant to the given activity (i,j) :



Start Node

Final Node

LST=latest Start node time

LFT=latest Final node time

EST=earliest Start node time

EFT=earliest Final node time

Definition of Slack:

The **total slack** in the activity is $9-(2+4)=3 = LFT - (EST+d)$

The **free slack** in the activity is $7-(2+4)=1 = EFT - (EST+d)$

The **independent slack** in the activity is $7-(3+4)=0 = EFT - (LST+d)$

Note that these are in *decreasing* order: this is always true, as $LFT > EFT$ and $LST > EST$.



Meaning of the Different Slacks

Total Slack, or Float The extra time that could be made available to an activity, assuming it has started at its earliest, without delaying the whole project time

$$\text{Total Slack} = LFT - (EST+d)$$

Free Slack, or Float The extra time that could be made available to an activity, assuming it has started at its earliest, without preventing beginning at the earliest *any* subsequent activity (i.e., not just the final time as with total slack)

$$\text{Free Slack} = EFT - (EST+d)$$

Independent Slack The extra time that could be made available to an activity, assuming it has started at its latest, without preventing beginning at the earliest any subsequent activity

$$\text{Independent Slack} = EFT - (LST+d)$$

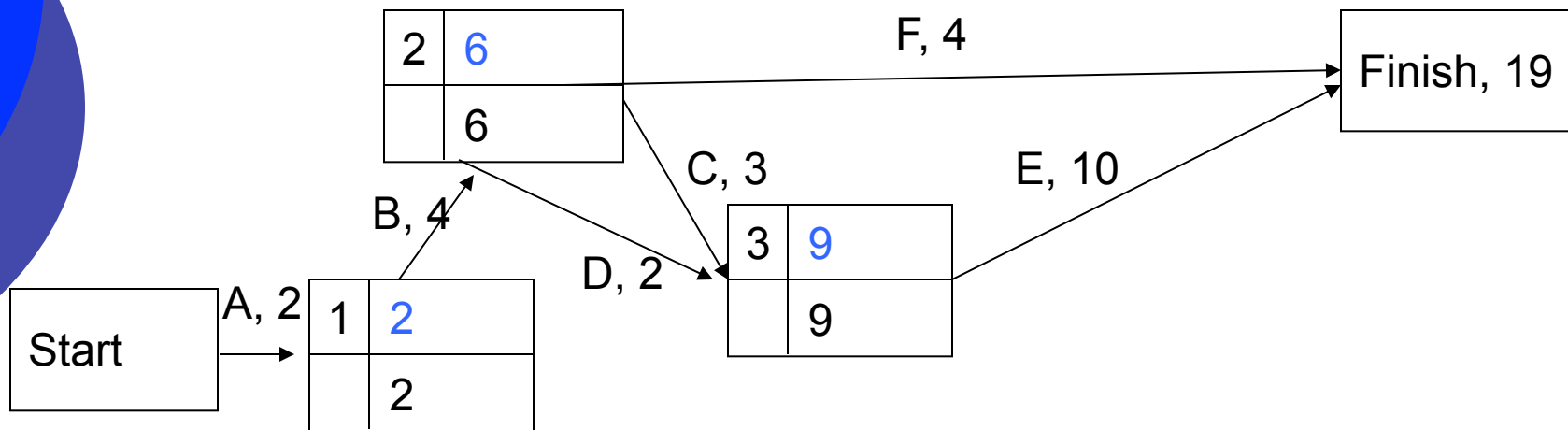
Any path through the network (from Start to Finish) with all zero total slack activities is a **critical path**.



Effect of Various Slacks in Decision-Making

- ⊕ If we use up the total slack in any activity, doing so may modify the critical path because of its influence on later activities. The coordinator of the whole project is therefore responsible for the total slack.
- ⊕ Free slack does not influence subsequent activities, but it may not be completely available, unless prior activities are started at their earliest.
- ⊕ Independent slack is a property of the activity alone: it does not require that any other activity be started at its earliest, and it does not affect the available durations of later activities. So this is the internal slack available to the person(s) responsible for that activity alone.

Example: Horizon Cable



Activity	A	B	C	D	E	F
Total Slack	0	0	0	1	0	9
Free Slack	0	0	0	1	0	9
Independent Slack	0	0	0	1	0	9
	Cr	Cr	Cr		Cr	

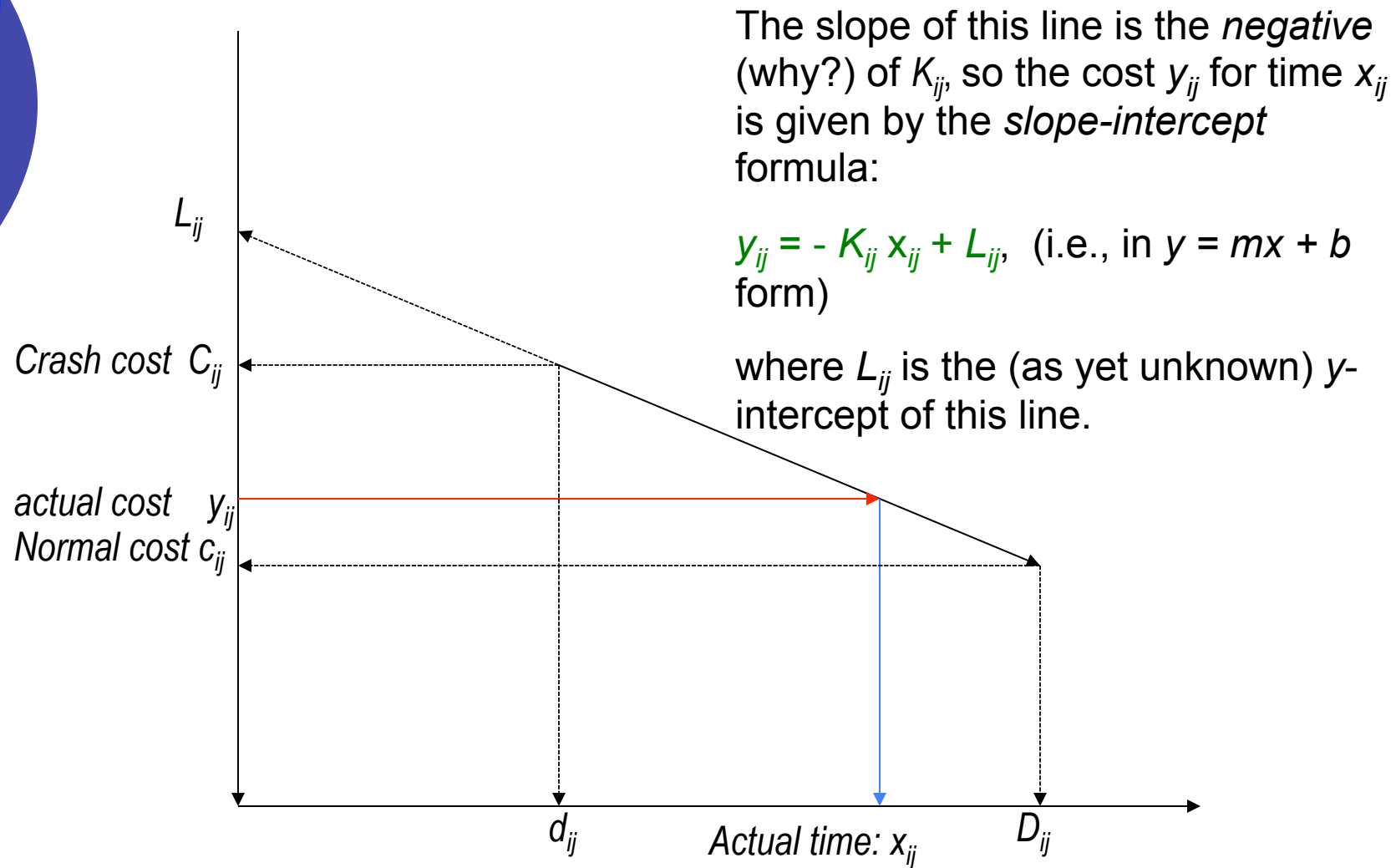
CPM: Crashing a Project

- ⊕ We use the following notation:
- ⊕ the normal time to complete an activity (i,j) is D_{ij} , and the normal cost of performing the activity in that time is c_{ij} .
- ⊕ The activity can be “crashed” by finishing it at a reduced time. The *minimum* time for the activity is d_{ij} ; for that time the (increased) cost, is C_{ij} .
- ⊕ In the Critical Path Method (CPM) the following is assumed:
- ⊕ *The relationship between activity (i,j) 's actual crash time, x_{ij} , to its cost is assumed to be linear.* We define the overall cost per unit time reduction, to be change in cost divided by change in time:

$$K_{ij} = (C_{ij} - c_{ij}) / (D_{ij} - d_{ij})$$

This gives us an equation for the cost y_{ij} for any crash time x_{ij} :

CPM: Crashing a Project



CPM: Crashing a Project

Thus the total cost is the sum of the costs of crashing along *all* activities:

$$\text{Cost} = \sum y_{ij} = - \sum K_{ij} x_{ij} + \sum L_{ij}$$

A final assumption is that

- ⊕ *The critical path does not change from that obtained by using the normal times, when some or all of the activities are crashed within their ranges.*

So we want to *minimize* the total cost, i.e., we want to minimize $-\sum K_{ij} x_{ij} + \sum L_{ij}$. But this is the same as *maximizing* its negative:

$$\text{Maximize } \sum K_{ij} x_{ij} - \sum L_{ij}$$

Finally, the sum of the y-intercepts, $\sum L_{ij}$, is a constant which does not affect the optimization process, so we can ignore this piece for now and all we actually need to do is to

CPM Theory Continued

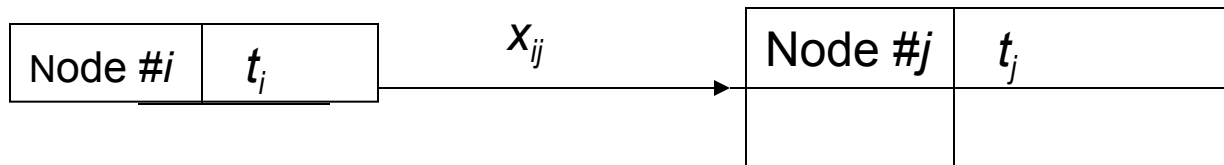
$$\text{Maximize } Z = \sum K_{ij} x_{ij}$$

This looks like it is becoming an LP problem and indeed it is: the *constraints* on this maximization are straightforward:

Along the critical path, we need to ensure that no activity starts before any prior activity has finished. So we introduce (variable) *node times* t_i for each node $\#i$ on the critical path. Then it is clear that node j cannot happen until at least activity (i,j) is done:

$$t_j \geq t_i + x_{ij} \text{ or } t_i \leq t_j - x_{ij}$$

(where by \leq we mean “less than or equal to” etc.) Also, each of the t_i is non-negative, and we need to ensure that the project is completed by the pre-determined (crashed) project time.





The CPM LP

Finally, each activity time has to be greater than or equal to the minimum, completely crashed time given. This is always at least zero, so we do not need non-negativity constraints! So we have formulated completely the CPM problem of time-cost tradeoffs:

In order to crash the project so as to achieve the stated project time, with minimal increase in cost, solve the following LP:

The CPM Linear Programming Problem

Variables: Activity times x_{ij} , Node completion times t_i , (including final node $t_{Final Node}$)

$$\text{Maximize } Z = \sum K_{ij} x_{ij}$$

Subject to:

$$t_i \leq t_j - x_{ij} \text{ for all critical activities } (i,j), \text{ and}$$

$$t_{Final Node} \leq \text{Desired Project Time}$$

$$x_{ij} \geq d_{ij}, \text{ the minimum time allowed for activity } (i,j)$$

$$x_{ij} \leq D_{ij}, \text{ the maximum time allowed for activity } (i,j)$$

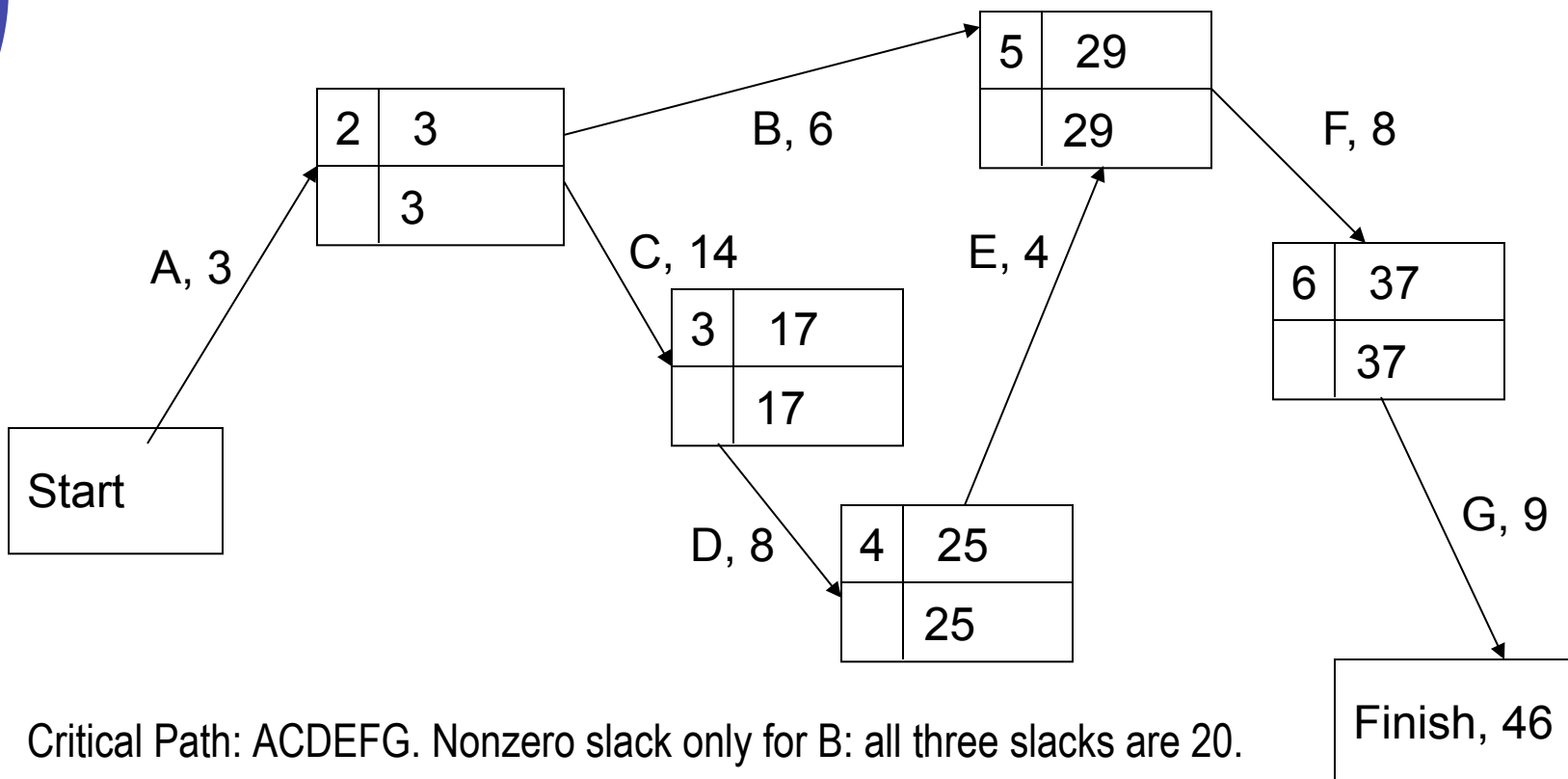
Example: Auditing a Corporation

⊕ When an accounting firm audits a corporation, the first phase of the audit involves obtaining "knowledge Of the business". This requires the activities in below. Draw the activity network, determine the critical path, and find the total float and free float for each activity.

Activity	A	B	C	D	E	F	G
Description	Determining terms of engagement	Appraisal of auditability risk	Identification of types of transactions and possible errors	Systems description	Verification of systems description	Evaluation of internal controls	Design of Audit approach
Immediate Predecessors	-	A	A	C	D	B,E	F
Duration (days)	3	6	14	8	4	8	9

Example for CPM: The Audit

The network is:



Critical Path: ACDEFG. Nonzero slack only for B: all three slacks are 20.
 Now let's see what happens when we have to crash the project:



Example: Crashing the Audit

Assume the auditing project must be completed in 30 days. The maximum allowable reduction in duration, as well as the cost per day of such a reduction, is appended below. Formulate an LP that can be used to minimize the cost of meeting the projected deadline.

Activity	A	B	C	D	E	F	G
Normal Duration (days): D_{ij}	3	6	14	8	4	8	9
Cost per day of reducing duration: K_{ij}	\$100	\$80	\$60	\$70	\$30	\$20	\$50
Maximum allowable reduction in duration	3	4	5	2	4	4	4
(Minimum time: d_{ij})	0	2	9	6	0	4	5

Example for CPM: The Audit

⊕ Subtracting the third row above from the second, we get the minimum times, d_{ij} . The cost per day of reduction, i.e., the K_{ij} , are given in the second column above. Finally, the critical path is ACDEFG, so the LP for completing the project by the deadline of 30 days is:

⊕ Maximize $Z = 100 x_A + 80 x_B + 60 x_C + 70 x_D + 30 x_E + 20 x_F + 50 x_G$

Subject to:

network time constraints along the critical path:

$$x_A - t_2 \leq 0; x_B + t_2 - t_5 \leq 0; x_C + t_2 - t_3 \leq 0;$$
$$x_D + t_3 - t_4 \leq 0; x_E + t_4 - t_5 \leq 0; x_F + t_5 - t_6 \leq 0; x_G + t_6 - t_7 \leq 0;$$

the total time constraint:

$$t_7 \leq 30;$$

minimum time constraints:

$$x_A \geq 0; x_B \geq 2; x_C \geq 9; x_D \geq 6; x_E \geq 0; x_F \geq 4; x_G \geq 5;$$

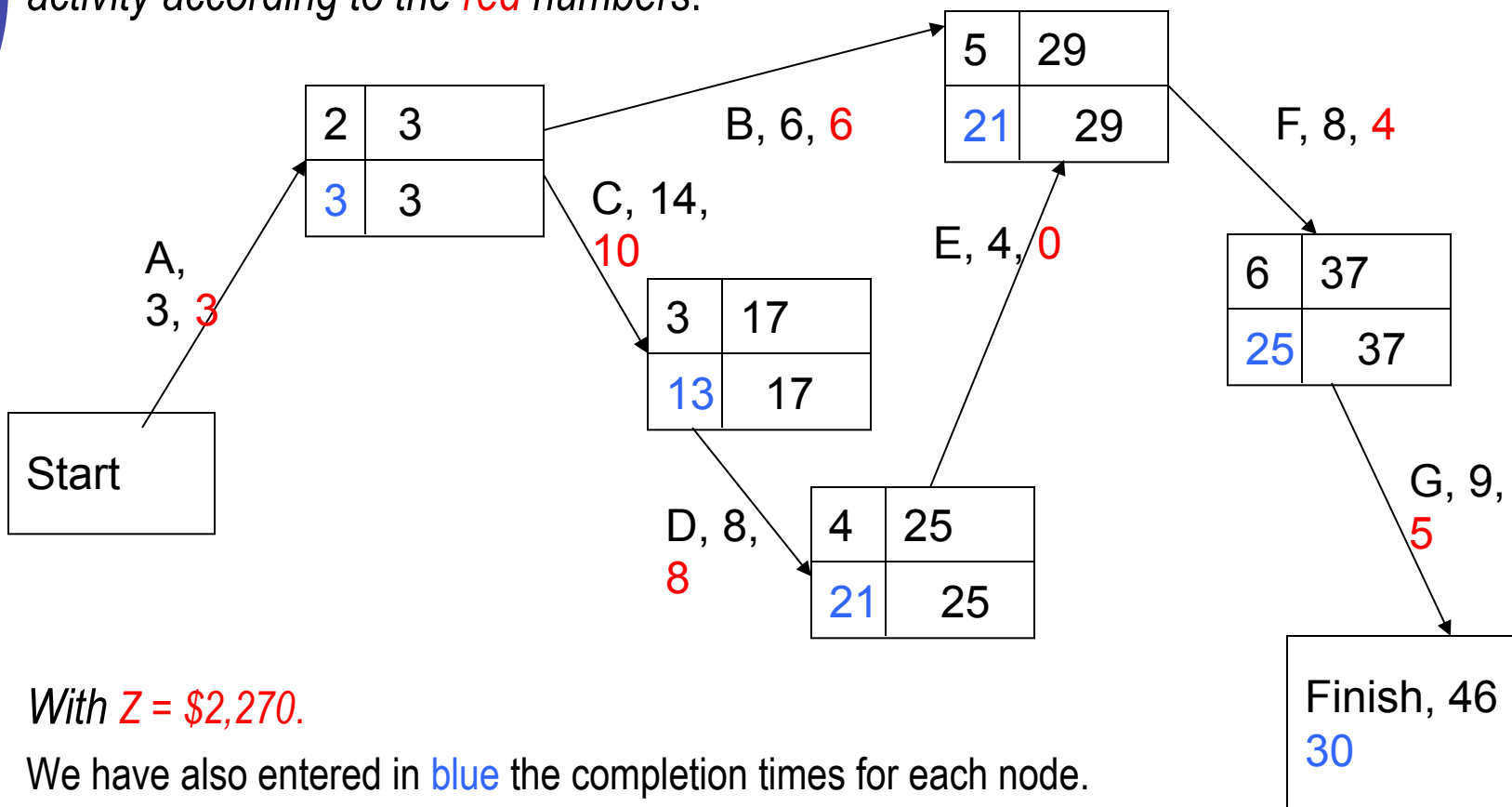
and maximum time constraints:

$$x_A \leq 3; x_B \leq 6; x_C \leq 14; x_D \leq 8; x_E \leq 4; x_F \leq 8; x_G \leq 9.$$

Note: It is important to include *all* the activities in Z , since this is a measure of *total* cost. Similarly, we must include *all* activities in the minimum-time constraints. Only the *network* constraints are restricted to the critical path.

Example for CPM: The Audit – Solution of the LP

Using MAPLE, the solution is: *In order to complete the project in 30 days, run each activity according to the red numbers:*



With $Z = \$2,270$.

We have also entered in blue the completion times for each node.



Calculating the Increased Cost

The quantity to be minimized, is the *total cost*, which is given by the formula on slide 31:
 $-\sum K_{ij} x_{ij} + \sum L_{ij} = \sum L_{ij} - Z$. Thus, if we can compute $\sum L_{ij}$ for all activities in the network, we will be able to find the minimum total cost.

⊕ For the line on slide 30, we know that the equation of the cost-duration line is $y_{ij} = -K_{ij} x_{ij} + L_{ij}$, and we know that the normal time-and-cost point (D_{ij}, c_{ij}) is on this line. Plugging that in and solving for L_{ij} , we see that

$$L_{ij} = c_{ij} + K_{ij} D_{ij}$$

Thus the total cost is $= \sum (c_{ij} + K_{ij} D_{ij}) - Z$

What does this mean?

Formula for Total Increased Cost; Meaning of Z

- ⊕ Total cost = $\Sigma (c_{ij} + K_{ij} D_{ij}) - Z = \Sigma c_{ij} + \Sigma K_{ij} D_{ij} - Z$
- ⊕ = Normal-Time Cost + (Cost/unit reduction)X(Normal Time) - Z
- ⊕ = Normal-Time Cost + Virtual Cost of reducing all times to zero - Z
- ⊕ Thus the **total increased cost** is $\Sigma K_{ij} D_{ij} - Z$, i.e., the
Virtual Cost of reducing all times to zero - Z
- ⊕ So Z turns out to be the *correction* we have to apply because some or all of the actual computed times are bigger than zero.
- ⊕ To find the increased cost in any actual example, introduce a new row which calculates the quantity $K_{ij} D_{ij}$ for each activity, add and finally subtract Z.
- ⊕ Another way to do this is to calculate each *actual* reduction in time, and multiply by the corresponding cost-per-unit reduction, and then add.

Example for CPM: The Audit – Solution of the LP

- ⊕ For this example, we had found $Z = \$3,390$. To the table of normal times, introduce a new row for the product (Cost/unit reduction) \times (Normal Time) and add along that row:

Activity	A	B	C	D	E	F	G
Normal Duration (days): D_{ij}	3	6	14	8	4	8	9
Cost per day of reducing duration: K_{ij}	\$100	\$80	\$60	\$70	\$30	\$20	\$50
Maximum allowable reduction in duration	3	4	5	2	4	4	4
(Minimum time: d_{ij}	0	2	9	6	0	4	5)
Row 1 X Row 2	300	480	840	560	120	160	450

So the first term is the sum along this last row: \$2,910. This is what it would have cost had we crashed every activity to zero (of course, this is disallowed by the minimum time requirements!). Subtracting $Z = \$2,270$, we get that the

Total minimal increased cost for the audit is \$640.

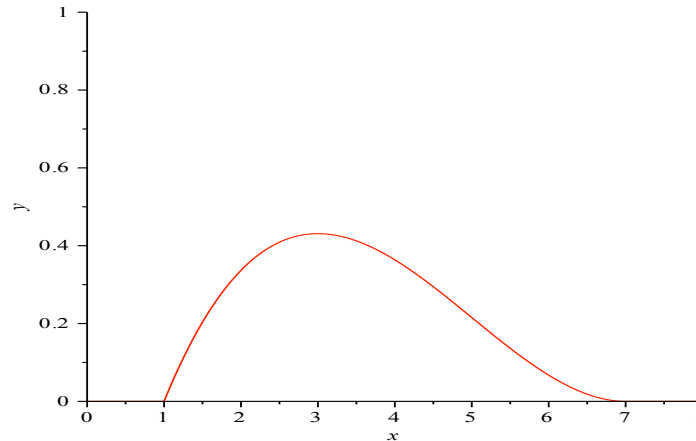
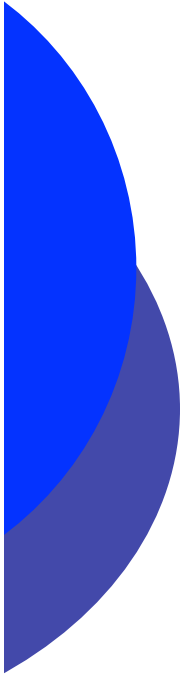


Program Evaluation Research Technique

- ⊕ In practice, activity times are somewhat random.
- ⊕ A good estimate for the distribution of an activity time can be made on the basis of past experience.
- ⊕ In some instances there is no past experience but a best guess at the distribution of activity times can still be made.

- ⊕ It turns out that, in general, the randomness of activity times is well approximated by the so-called “beta” distribution. This is a distribution that is completely determined by three parameters: the shortest, or optimistic time a , the longest, or pessimistic time b and the mode, or most likely time m .
- ⊕ In the following we will show how knowing these three numbers for each activity will allow an estimate of the likelihood of meeting a specific completion date.
- ⊕ This is therefore also called the “three estimate approach.”

The Beta Distribution



Here is a typical Beta distribution, with $a = 1$, $b = 7$ and m approximately 3. (Why is the shape of the beta distribution (i.e., skewed to the right) reasonable for activity times?)

⊕ Fact: For any beta distribution with parameters a , m and b , the mean is:

$$t = (a + 4m + b)/6$$

And the variance (i.e., *square* of the standard deviation) is

$$V = (b-a)^2/36$$



Sums of Independent Random Variables

- ⊕ Here's another fact from probability theory:

Sums of Independent Random Variables

- ⊕ If X_1, X_2, \dots, X_n are *independent* random variables with means $\mu_1, \mu_2, \dots, \mu_n$ and variances V_1, V_2, \dots, V_n respectively, then for the sum random variable

$$X = X_1 + X_2 + \dots + X_n,$$

the mean is $\mu_1 + \mu_2 + \dots + \mu_n$ and the variance is $V_1 + V_2 + \dots + V_n$:

Given independence, the *means and variances add*

Note: The standard deviations do not add to give the total standard deviation! We have to first add their squares (i.e., the variances), and then take the square root, to find the “total” standard deviation.

So:

The total project time is a random variable, that has a mean and a variance that are both *sums of the same along the critical path*



Central Limit Theorem

- ⊕ Here's another fact from probability theory:

Central Limit Theorem

The sum of a (sufficiently large) number of independent random variables is approximately normally distributed

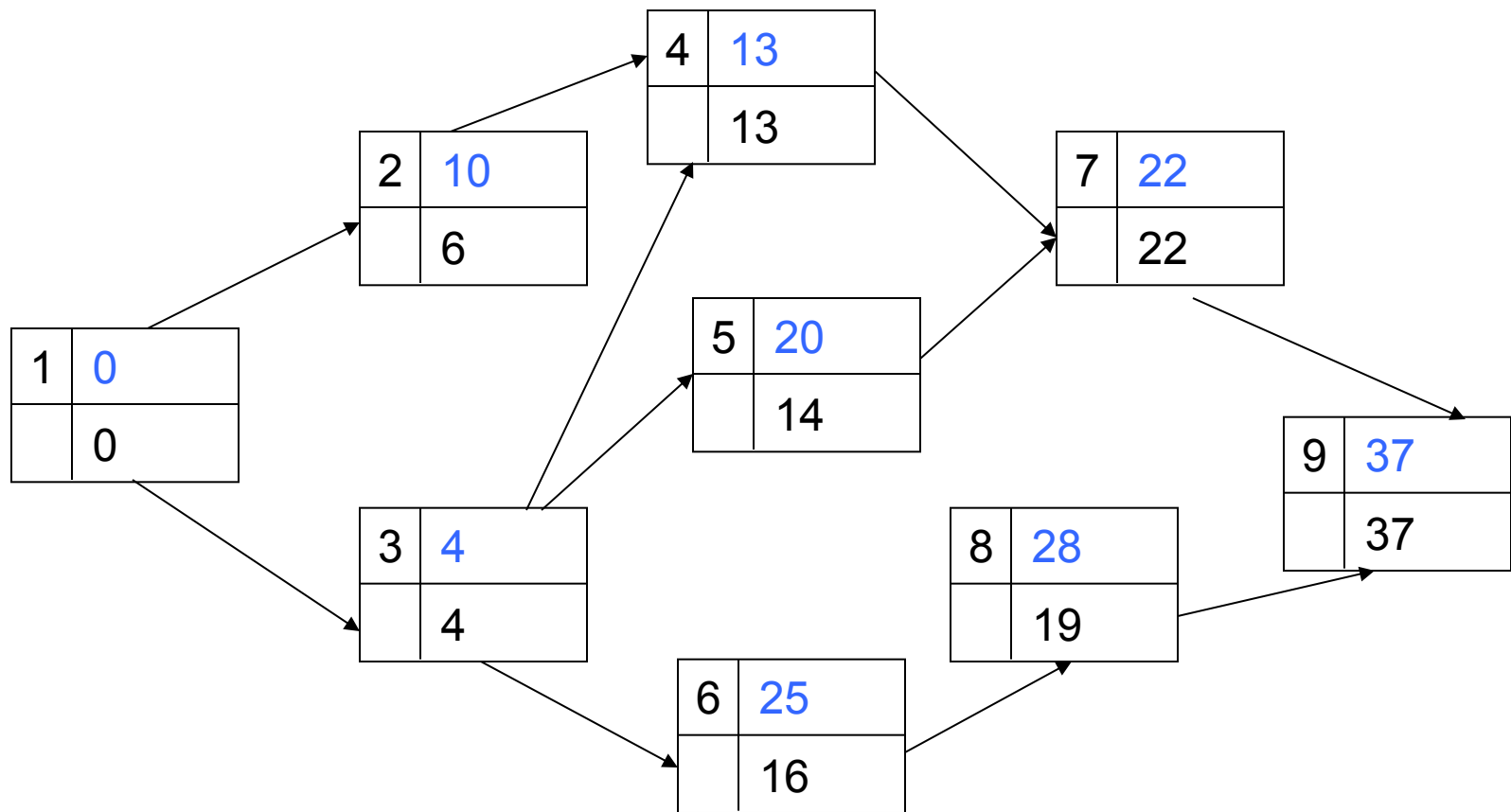
- ⊕ We are assuming that the critical path does not change in the range (a to b) of activity times given by the parameters for the set of activities
- ⊕ We find the critical path by using the modal times m for the activities.
- ⊕ Thus the project completion time random variable will have a **normal distribution** with *mean equal to the sum of the means along the critical path* and *variance equal to the sum of the variances along the critical path*.



PERT Method

- 1 Draw the activity network, as in CPM
- 2 Work out the critical path, using the modal times m for each activity, as in CPM
- 3 Find and then add the means only along the critical path, using the formula
$$\mu = \sum (a + 4m + b)/6$$
- 4 Find and then add the Variances only along the critical path, using the formula
$$V = \sum (b-a)^2/36$$
- 5 Find σ = the square root of V
- 6 The total time X along the critical path is now a *normally distributed* random variable with mean μ (the *expected project time*) and standard deviation σ .
- 7 Find the probability that the random total project time variable X is less than, or equal to, any given specified completion time T , i.e., $P(X \leq T)$ by converting to Z-scores and using the standard normal distribution tables.

Example: #7 on the Handout



Example: #7 on the Handout

- ⊕ The above is the full network with times computed using the m 's from the table given in the problem, giving a total time of 37. Here it is clear that the critical path is

1-3-4-7-9

and that the floats (slacks) are:

Activity	(1,2)	(1,3)	(2,4)	(3,4)	(3,5)	(3,6)	(4,7)	(5,7)	(6,8)	(7,9)	(8,9)
Total Float	4	0	4	0	6	9	0	6	9	0	9
Free Float	0	0	4	0	0	0	0	6	0	0	9
Independent Float	0	0	0	0	0	0	0	0	0	0	0



Example: #7 on the Handout

- ⊕ We next compute the means and variances of each of the activities along the critical path 1-3-4-7-9:

Activity	(1,3)	(3,4)	(4,7)	(7,9)	Sum
Mean	6	9	8.83	15	38.83
Variance	1	1	1.36	2.78	6.14

- ⊕ Notice that the mean along the critical path is slightly greater than the length computed using the m 's. This is reasonable for a distribution that is skewed to the right.
- ⊕ Thus the project time is $N(38.83, 2.48)$ (the standard deviation is the square root of the variance of 6.14).



Example: #7 on the Handout

- ⊕ Thus the probability that this normal random variable is less than or equal to 40 is given by the probability that a standard normal random variable is less than or equal to

$$\oplus Z = (40 - 38.83)/2.48 = 0.47$$

- ⊕ From the tables, this is 0.6808:

- ⊕ Answer: **The probability that the project will be completed in 40 days is 68%**